International Journal of Exploring Emerging Trends in Engineering (IJEETE)
Vol. 03, Issue 02, Mar-Apr, 2016 Pg.72-78 WWW.IJEETE.COM

DESIGN AND IMPLEMENTATION OF AN ERROR LESS AND EFFICIENT TRANSMISSION AMONG NANO SCALE DEVICES

[*]KOTTNANA JANAKI RAM, [**]BORRA ANUSHA SAMRAT
[*]Assistant Professor, M.Tech (VLSI & ES). [**]Assistant Professor, M.Tech (CE&SP).
ECE Department, Viganan's Nirula Institute of Technology & Science for women.

Abstract:

This project mainly introduced for decreasing soft errors which occurs around the Nano memory. Due to the increase in soft error rate in logic circuits, the encoder as well as decoder circuitry around the memory blocks have become susceptible to soft errors as well and must also be protected. So we are introducing a new approach to design fault-secure encoder and decoder circuitry for memory designs. Thermal along with shot-noise estimations alone suggest that the transient fault rate of an individual Nano scale mechanism (e.g., transistor or else Nano wire) may be orders of magnitude higher than today's devices. As a result, we can look ahead to combinational logic to be susceptible to transient faults in addition to storage cells and communication channels. Consequently, the paradigm of protecting only memory cells and assuming the surrounding circuitries (i.e., encoder and decoder) will never initiate errors is no longer valid.

We provide the design structure of the encoder, corrector, along with detector units of our proposed fault-tolerant memory structure. Fault secure detectors detect the errors in encoders by generating the syndrome vectors with the help of parity check matrix. Coming to the decoder side correction majority logic correction is the course of action that identifies the correct value of a each bit in the codeword directly from the received codeword.

Index Terms: EG-LDPC Algorithm, Majority logic decoding and encoding, Nano scale devices, Syndrome Computations.

I.INTRODUCTION

Nano technology provides slighter, faster, and lower energy devices which allow more powerful and compact circuitry; on the other hand, these benefits come with a cost—the Nano scale devices may be less reliable.

A new move toward to design fault-secure encoder as well as decoder circuitry for memory designs is introduced. key novel contribution of this project is identifying furthermore defining a new class of error-correcting codes whose redundancy makes the intend of fault-secure detectors (FSD) particularly straightforward. LDPC codes satisfies a innovative, restricted definition for ECCs which guarantees that the ECC codeword has an suitable redundancy structure such that it can detect multiple errors occurring in together the stored codeword in memory along with the surrounding circuitries.

II.EXISTING METHOD

This paper introduces a fault-tolerant nano scale memory architecture which bear transient faults in cooperation in the storage unit and in the supporting logic (i.e., encoder, decoder (corrector), with detector circuitries)[1] these codes can correct a larger number of errors, except generally require complex decoders. To avoid a high decoding complexity, the utilize of one step majority logic decodable codes was first proposed in [4] for memory applications. Further effort on this topic was then presented in [5], [4], [2]. One step majority logic decoding can be implemented serially with very easy circuitry [9], but requires long decoding times. In a memory, this would increase the admittance time which is an important system parameter. Only a few classes of codes can be decoded using one step majority logic decoding [5]. Among those are a number of Euclidean geometry low density parity check (EG-LDPC) codes which were used in [4], moreover difference set low density parity check (DS-LDPC) codes [9].

A) Architecture of ECC Nano Memory

Figure1 shows the overall system architecture of the nano memory among EG- LDPC ECC. During a write process, the inward word to be stored in memory is encoded by the encoder and the code word is stored in memory. During a read process, a code word is retrieved from the memory, checked by the checker unit, and finally the majority logic unit decodes the syndrome furthermore does the error correction. The controller unit controls the error detection along with correction capability of the ECC unit. The following sections describe these different components and their implementation using nano-PLA components.

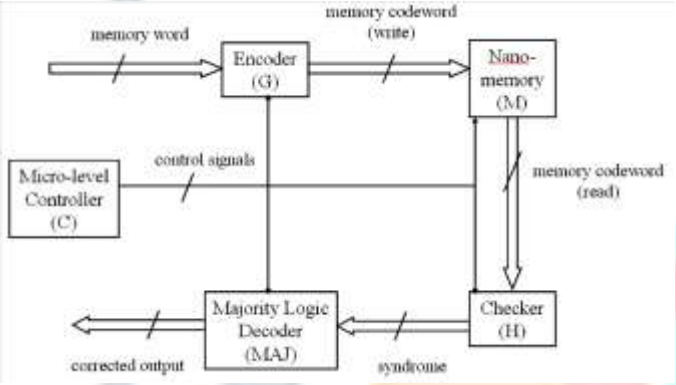


Figure:1 Overall architecture of ECC nano memory

III.PROPOSED METHOD

A technique was recently proposed in [1] to accelerate a serial implementation of majority logic decoding of DS-LDPC codes. The thought behind the technique is to use the first iterations of majority logic decoding to detect if the word being decoded contains errors. If there are no errors, subsequently decoding can be stopped without completing the remaining iterations, therefore greatly reducing the decoding time.

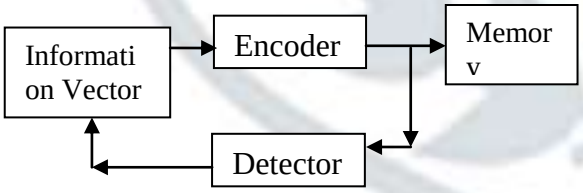


Figure2: Error checking in encoder

The impact of technology scaling smaller dimensions, higher integration densities, as well as lower operating voltages has come to a level that reliability of memories is put into jeopardy, not only in tremendous radiation environments like spacecraft and avionics electronics, but also at normal terrestrial environments[1][2],.

Mitigation techniques are:

- Triple modular redundancy (TMR)
- Error correction codes (ECCs).

TMR is a particular case of the von Neumann method consisting of three versions of the design in parallel, with a majority voter selecting the accurate output as the method suggests, the complexity overhead would be three times plus the complexity of the majority voter furthermore thus increasing the power consumption. For memories, it turned out that ECC codes are the finest way to mitigate memory soft errors For terrestrial radiation environments where there is a low soft error rate (SER),[2] codes like single error correction and double error detection (SEC–DED)[2], are a superior solution, due to their low encoding and decoding complexity. However, as a consequence of augmenting integration densities, there is an increase in the number of soft errors, which produces the need for higher error correction capabilities The usual multi error corection codes,suchas[1]Reedsolomon(RS) or (BoseChaud-huri–Hocquenghem) BCH[5] are not suitable for this task. The basis for this is that they use more sophisticated decoding algorithms, like complex algebraic decoders that can decode in fixed time, as well as simple graph decoders, that use iterative algorithms Both are very complex and increase computational costs

For a code with block length N, majority logic decoding (when implemented serially) requires N iterations, so that as the code volume grows, so does the decoding time. In the proposed approach, only the first three iterations are used to detect errors, by this means achieving a large speed increase when N is large. In [1] it was shown that for DS-LDPC codes, all error groupings of up to five errors can be detected in the first three iterations. An extra advantage of the proposed method is that it requires very little additional circuitry as the decoding circuitry is also used for error detection.

N	K	J	t_{ML}
15	7	4	2
63	37	8	4
255	175	16	8

1023	781	32	16

Table1: One Step MLD EG-LDPC Codes

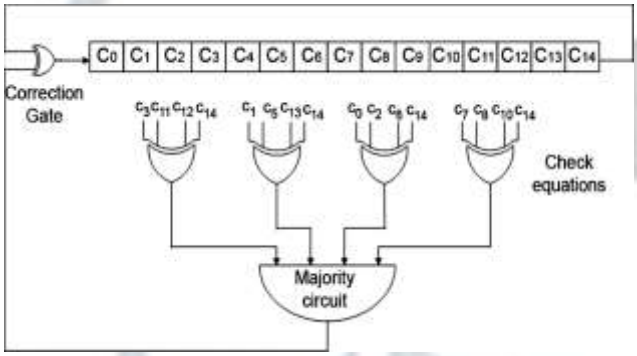


Figure 3: Serial one-step majority logic decoder for the EG-LPDC code (Decoder corrector)

for the coded vectors C_0,C_1,C_2,C_3,C_4,C_5,C_6 directly inputs are given. for encoded vectors C_7,C_8,C_9,C_{10},C_{11},C_{12},C_{13}.C_{14} these encoder inputs are linearly ex-ored with given to the output .finally ,the 'x'matrix is derived.

Detection of errors in decoder:

Detection of errors in decoder One-step mainstream logic correction is the procedure that identifies the correct value of a all bit in the codeword directly from the received codeword this is in contrast to the general message-passing error correction strategy This technique consists of two parts:

1) Generating a specific set of linear sums of the received vector bits moreover

2) Finding the majority significance of the computed linear sums

From above every matrixes SYNDROME matrix is calculated. If Syndrome (s) vector contains all 'zero's afterward the received code word is correct. If it contains some 'one', received code word is wrong if any error occurred after that immediately detector once additional intimate to encoder to produce correct coded word. If no error found then it just ignores it.

Finite geometries have been used to obtain many error-correcting codes[5],[1]. One example is EG-LDPC codes which are based on the configuration of Euclidean geometries over a Galois field. In the midst of EG-LDPC codes there is a subclass of codes that is one step majority logic decodable (MLD) [5]. Codes in this subclass are also cyclic. The parameters for some of these codes are given in Table I, where N is the block size, K the number of information bits, J the number of MLD check equations and t_{ML} the figure of errors that the code can correct using one step MLD.

One step MLD can be implemented serially using the scheme in Fig. 1 which corresponds to the decoder for the EG-LDPC code with N=15. Primarily the data block is loaded into the registers. Then the check equations are computed furthermore if a majority of them has a value of one, the last bit is inverted. Then all bits are cyclically shifted. This set of operations constitutes a particular iteration: after N iterations, the bits are in the same position in which they were loaded. In the process, every one bit may be corrected only once. As can be seen, the decoding circuitry is simple, but it requires a long decoding time if N is large. The check equations must have the following properties (see [5] for more details)

Table2: Undetected Errors in Exhaustive Checking

N	1 error	2 errors	3 errors	4errors
15	0	0	0	0
63	0	0	0	0
255	0	0	0	--
1023	0	0	---	--

1) All equations include the variable whose value is stored in the last register (the one marked as C_{14}).

2) The rest of the registers are included in at most one of the check equations.

If errors can be detected in the foremost few iterations of MLD, then whenever no errors are detected in those iterations, the decoding can be stopped without completing the rest of the iterations. In the first iteration, errors will be detected when at least one of the make sure equations is affected by an odd number of bits in error. In the second iteration, as bits are cyclically shifted by one position, errors will affect additional equations such that some errors undetected in the first iteration will be detected. As iterations move forward, all detectable errors will eventually be detected.

In [1] it was shown that for DS-LDPC codes most errors can be detected in the first three iterations of

MLD. Based on simulation results in addition to on a theoretical evidence for the case of two errors, the following hypothesis was made. "Given a word read from a memory protected with DS-LDPC codes, and affected by up to five bit-flips, all errors can be detected in merely three decoding cycles". Then the proposed technique was implemented in VHDL along with synthesized, showing that for codes with large block sizes the overhead is small. This is because the existing majority logic decoding circuitry is reused to perform error detection and only some extra control logic is needed.

The remainder of this paper is organized as follows. Section II gives an general idea of existing ML decoding so-lutions; Section III presents the novel ML detector/decoder (MLDD) by means of difference-set cyclic codes; Section IV dis- cusses the results obtained for the dissimilar versions in respect to effectiveness, performance, and area and power consump- tion. Finally, Section V discusses conclusions and gives an outlook onto future work.

Figure4:Memory with MLD

IV.MAJORITY LOGIC DECODING (MLD)

An n-bit code-word C, which encodes k-bit information vector i is generated by multiplying the k-bit information vector with k × n bit generator matrix G, i.e., c = i · G. Figure 5 shows the generator matrix of (15, 7) EG-LDPC code. all the rows of the matrix are cyclic shifts of the first row. This cyclic code generation does not produce a systematic code in addition to the information bits must be decoded from the encoded vector, which is not desirable for our fault-tolerant approach due to the additional complication and delay that it adds to the operation.
The generator matrix of any cyclic code can be converted into systematic form (G = [I : X])

	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}
i_0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0
i_1	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0
i_2	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0
i_3	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0
i_4	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0
i_5	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0
i_6	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1

Figure 5: The generator matrix of EG-LDPC code of (15, 7) in cyclic format

Figure 5 shows the systematic generator matrix to generate (15, 7) EG-LDPC code[2]. The encoded vector, which is generated by the inner product of the information vector and the generator matrix, consists of information bits followed by parity bits, where each parity bit is simply an inner product of information vector and a column of X, from G = [I : X].

The encoder circuit to compute the parity bits of the (15, 7) EG-LDPC code. In this figure i = (i_0, ..., i_6) is the information vector and will be copied to c_0, ..., c_6 bits of the encoded vector, c, and the rest of encoded vector, the parity bits, are linear sums (Ex-or) of the information bits. If the building block is two-input gates then the encoder circuitry takes 22 two input Ex-or gate. Since the systematic generator matrix of EG-LDPC and PG-LDPC codes does not have the standard row and column density, To compute the area of an encoder circuitry the corresponding systematic generator matrix has to be constructed. Once the systematic generator matrix is constructed the fan_{in} size of the Ex-or gates can be determined by the column densities of the generator matrix.

One-step majority-logic rectification is a fast and relatively compact error-correcting technique. There is a limited class of ECCs that are one-step-majority correctable which include type-I two-dimensional EG-LDPC. In this section, we present a brief review of this correcting technique. Then we show the one-step majority-logic corrector for EG-LDPC codes.

1) One-Step Majority-Logic Corrector: One-step majority logic correction is the procedure that identifies the correct value of a every bit in the codeword directly from the received codeword; this is in distinction to the general message-passing error correction strategy, which may demand multiple iterations of error diagnosis and trial correction. Avoiding iteration makes the correction latency both small and deterministic. This technique can be implemented serially to provide a compact implementation or in parallel to minimize correction latency.
This method consists of two parts:

1) generating a specific set of linear sums of the received vector bits furthermore

2) finding the majority value of the computed linear sums. The majority value indicates the correctness of the code-bit beneath consideration; if the majority value is 1, the bit is inverted, otherwise it is kept unchanged.

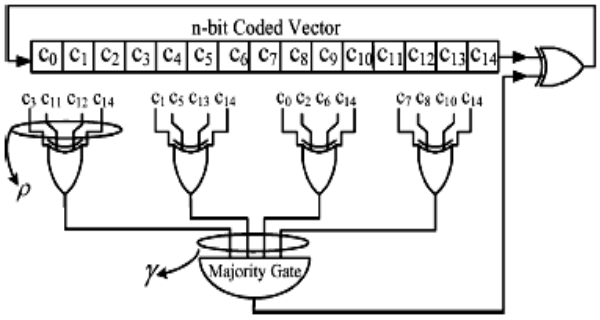


Figure 6: Serial one-step majority logic corrector structure

A linear sum of the received encoded vector bits can be formed by computing the inner product of the received vector and a row of a parity-check matrix. This sum is called Parity-Check sum. A set of parity-check sums is said to be orthogonal on a given code bit if each of the parity-check sums include the code bit but no other code bit is included in more than one of these parity-check sums. If for each code bit there are j parity-check sums that are orthogonal on it, then the code is one-step majority-logic correctable up to bj/2c bit errors. In a cyclic code, a set of j parity-check sums orthogonal on a code-word bit is orthogonal on all the n code-word bits. Therefore, using one set of parity-check matrix rows orthogonal on one code bit, we can design a majority circuit that corrects all the other bits, serially.

The one-step majority logic error correction is summarized in the following procedure.

These steps correct a potential error in one code bit, e.g., c_n-1.

1. The j parity-check sums orthogonal on c_n-1 are formed by computing the inner product of the received vector and the appropriate rows of parity-check matrix.

2. The J orthogonal check sums are fed into a majority gate. The output of the majority gate corrects the bit c_n-1, by inverting the value of c_n-1 if the output of majority gate is "1".

The circuit implementing a serial one-step majority logic corrector for (15, 7) EG-LDPC code is shown in figure 2 is circuit generates parity-check sums with Ex-or gates and then computes, the majority value of the parity-check sums. Since each parity-check sum is computed using a row of the parity-check matrix and the row density of EG-LDPC codes are _ then each ex-or gate that computes the linear sum has _ inputs. The single Ex-or gate on the right, corrects the code bit c_n-1, using the output of the majority gate. Once the code bit c_n-1 is corrected the code-word is cyclic shifted and code bit c_n-2 is placed at c_n-1 position and will be corrected. The whole code-word can be corrected in n rounds.

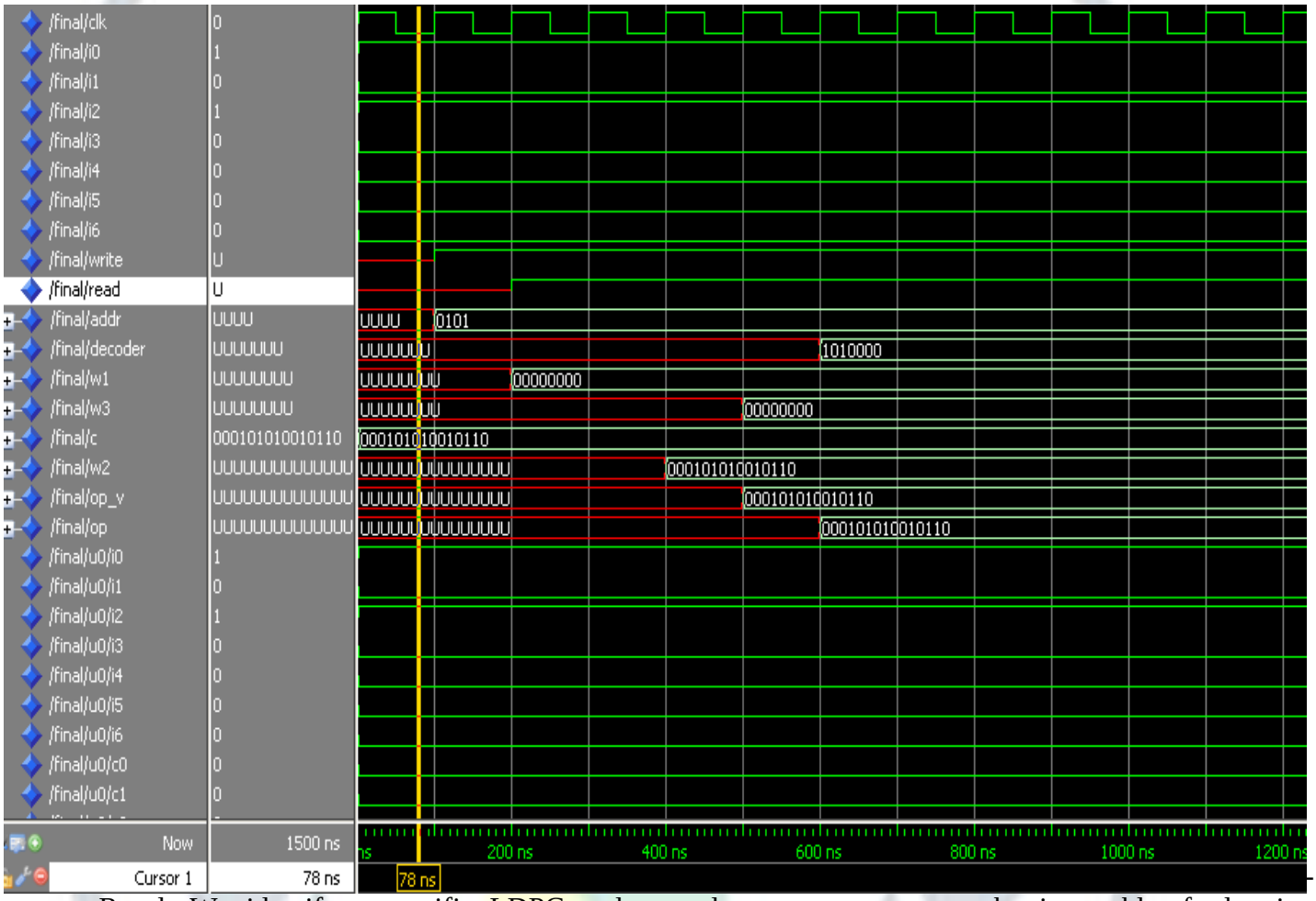
The method proposed in [1] has been applied to the class of one step MLD EG-LDPC codes. To present the results, the conclusions are presented first in terms of a hypothesis that is then validated by simulation and also partially by a theoretical analysis presented in the Appendix. The results obtained can be summarized in the following hypothesis. "Given a word read from a memory protected with one step MLD EG-LDPC codes, and affected by up to four bit-flips, all errors can be detected in only three decoding cycles". Note that this hypothesis is different from the one made for DS-LDPCs codes in [1] as in that case errors affecting up to five bits were always detected. This is due to structural differences between DS-LDPC and EG-LDPC codes, which will be detailed in the Appendix. To validate the above hypothesis, the EG-LDPC codes considered have been implemented and tested. For codes with small words and affected by a small number of bit flips, it is practical to generate and check all possible error combinations.

As the code size grows and the number of bit flips increases, it is no longer feasible to exhaustively test all possible combinations. Therefore the simulations are done in two ways, by exhaustively checking all error combinations when it is feasible and by checking randomly generated combinations in the rest of the cases.

V. IMPLEMENTATION AND RESULTS

The proposed Memory testing using MLD is designed using Verilog hard ware description language and structural form of coding. The basic block of paper is Ex-or matrix, majority gates and cyclic shift registers. The design is completely

synchronized by the clock. The code is completely synthesized using Xilinx XST and implemented on device family Spartan 3e, device XC3S50OE, package FG320 with speed grade -4.



Result We identify a specific LDPC code that can tolerate up to 33 errors in each memory word or supporting logic that requires only 30% area overhead for memory blocks of 10 Kbits or larger. Larger codes can achieve even higher reliability and lower area overhead. We quantify the importance of protecting encoder and decoder (corrector) circuitry and illustrate a scenario where the system failure rate (FIT) is dominated by the failure rate of the encoder and decoder.

We are mainly used in BIST Applications and nano memory application, mainly checking the embedded Hardware boards by using BIST and EG-LDPC code techniques

VI.CONCLUSION

tolerant memory system that is capable of tolerating errors not only in the memory bits but also in the supporting logic including the ECC encoder and corrector. We used Euclidean Geometry codes. We proved that these codes are part of a new subset of ECCs that have FSDs. Using these FSDs we design a fault tolerant encoder and corrector, where the fault-secure detector monitors their operation. We also presented a unified approach to tolerate permanent defects and transient faults. This unified approach reduces the area overhead.

Feature enhancement in this project the power consumed is very high. So, in order to reduce power we can use Low transitions random test generation along with this method

REFERENCES

[1] pedro reviriego, juan a. maestro, and mark f. flanagan ,Error detection in majority logic decoding of Euclidean geometry low density parity check (eg-ldpc) codes ieee transactions on very large scale integration (vlsi) systems, vol. 21, no. 1, january 2013
[2] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," IEEE Trans. Device Mater. Reliab., vol. 5, no. 3, pp. 301–316, Sep. 2005.
[3] M. A. Bajura, Y. Boulghassoul, R. Naseer, S. DasGupta, A. F.Witulski, J. Sondeen, S. D. Stansberry, J. Draper, L. W. Massengill, and J. N. Damoulakis, "Models and algorithmic limits for an ECC-based approach to hardening sub-100-nm SRAMs," IEEE Trans. Nucl. Sci., vol. 54, no. 4, pp. 935–945, Aug. 2007.
[4] R. Naseer and J. Draper, "DEC ECC design to improve memory reliability in sub-100 nm technologies," Proc. IEEE ICECS, pp. 586–589, 2008.
[5] S. Ghosh and P. D. Lincoln, "Dynamic low-density parity check codes for fault-tolerant nano-scale memory," presented at the Foundations Nanosci. (FNANO), Snowbird, Utah, 2007.

AUTHOR'S BIBLOGRAPGHY

Mr. K.Janaki Ram Received M.Tech Degree in VLSI & ES of Engineering from MIC College of Technology in 2014.Presently working as Assistant Professor of the department of ECE in Vignans Nirula Institute of Technology for Women ,Guntur, India.

Miss. B.Anusha Samrat Received M.Tech degree in CE & SP of Engineering from Bapatlla engineering college in 2014. she is currently working as Assistant Professor the department of ECE in Vignans nirula institute of technology for women ,Guntur, India.

